**Amendments to the Claims**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of the claims:**

Claim 1.        (original)  A Java run-time system comprising

a stacked-based interpreter for executing a Java program comprising Java bytecode instructions and Java class structures;

a converter for mapping standard Java symbolic linking strings contained in a downloaded Java program onto linking identifiers; and

an export table for storing linking identifiers generated by the converter to bind a reference in a bytecode instruction to be executed to a corresponding link target.

Claim 2.        (original)  A Java run-time system according to claim 1, wherein the converter is adapted to use a hash function to map the standard Java symbolic linking strings onto linking identifiers.

Claim 3.        (original)  A Java run-time system according to claim 1, wherein the converter is adapted to use a parameterized hash function to map the standard Java symbolic linking strings onto linking identifiers, the parameter being contained in the Java program to be downloaded.

Claim 4.        (currently amended)  A ~~Java development kit comprising a~~ Java run-time system according to claim 3 ~~and~~ further comprising a Java conversion system for calculating a parameter for standard Java symbolic linking strings of a Java program to be downloaded on the Java run-time system.

Claim 5.        (original)  A Java run-time system according to claim 1, said run-time system being ported on an embedded microcontroller of a smart card.

Claim 6.        (original)  A Java run-time system according to claim 2, said run-time system being ported on an embedded microcontroller of a smart card.

Claim 7.        (original)  A Java run-time system according to claim 3, said run-time system being ported on an embedded microcontroller of a smart card.

Claim 8. (currently amended) A method for downloading and linking a Java program on a Java run-time system comprising a stack-based interpreter for executing bytecode instructions, said Java program comprising Java bytecode instructions and Java class structures and including Java standard symbolic linking strings to bind a reference in a bytecode instruction to be executed to a corresponding link target, comprising the steps of mapping the Java standard symbolic linking strings to linking identifiers; and storing said linking identifiers in an export table; and wherein the parameter for the Java program to be downloaded is used to ensure that the hash function does not map two symbolic linking strings of Java program to the same linking identifier .

Claim 9. (original) A method according to claim 8, wherein a hash function is used to map the standard Java symbolic link strings onto linking identifiers.

Claim 10. (original) A method according to claim 8, wherein a parameterized hash function is used to map standard Java symbolic linking strings onto linking identifiers, said parameter being included in the Java program to be downloaded.

Claim 11. (cancelled)

Claim 12. (original) A program storage device readable by a computer, tangibly embodying a Claim program of instructions executable by the computer to perform method steps of a method for downloading and linking a Java program on a Java run-time system comprising a stack-based interpreter for executing bytecode instructions, said Java program comprising Java bytecode instructions and Java class structures and including Java standard symbolic linking strings to bind a reference in a bytecode instruction to be executed to a corresponding link target, comprising the steps of mapping the Java standard symbolic linking strings to linking identifiers; and storing said linking identifiers in an export table.

Claim 13. (original) A program storage device according to claim 12, wherein a hash function is used to map the standard Java symbolic link strings onto linking identifiers.

Claim 14. (original) A program storage device according to claim 12, wherein a parameterized hash function is used to map standard Java symbolic linking strings onto linking identifiers, said parameter being included in the Java program to be downloaded.

Claim 15. (original) A program storage device according to claim 14, wherein the

parameter for the Java program to be downloaded is used to ensure that the hash function does not map two symbolic linking strings of Java program to the same linking identifier.

Claim 16.    (original)   A program storage device according to claim 12, said computer comprising an embedded microcontroller of a smart card.